Google

Web Images Groups News Froogle Local[New!] more »

how to share a secret shamir Search Advanced Search Preferences

The following words are very common and were not included in your search: how to a. [details]

Web Results **1** - **10** of about **36,000** for **how to share a secret shamir**. (**0.12** seconds)

Shamir's Secret Sharing
Nick Szabo's Papers and Concise Tutorials How to **Share** a **Secret**. Adi **Shamir** Massachusetts Institute of Technology In this paper we ...
szabo.best.vwh.net/**secret**.html - 11k - Cached - Similar pages

Citations: How to **Share** a **Secret** - **Shamir** (ResearchIndex)
... A. **Shamir**, How to **Share** a **Secret**, Communications of the ACM, vol. 22, no. ... A. **Shamir**, How to **share** a **secret**, Communications of the ACM, vol. 22, n.11, pp. ...
citeseer.ist.psu.edu/context/7538/0 - 15k - Cached - Similar pages

Publicly Verifiable **Secret Sharing** - Stadler (ResearchIndex)
... **Sharing** - Kulesza, Kotulski, Pieprzyk (2003) (Correct) Related documents from co-citation: More All 18: How to **Share** a **Secret** (context) - **Shamir** - 1979 16: How ...
citeseer.ist.psu.edu/stadler96publicly.html - 22k - Cached - Similar pages
[More results from citeseer.ist.psu.edu]

[PDF] How to **Share** a **Secret**
File Format: PDF/Adobe Acrobat - View as HTML
Page 1. Programming R. Rivest Techniques Editor How to **Share** a **Secret** Adi **Shamir** Massachusetts Institute of Technology In this paper ...
crypto.csail.mit.edu/classes/ 6.857/papers/**secret-shamir**.pdf - Similar pages

Secret sharing - Wikipedia, the free encyclopedia
... Adi **Shamir**, "How to **share** a **secret**", Communications of the ACM, 22(1), pp612–613, 1979 [1] (http://www.cs.tau.ac.il/~bchor/**Shamir**.html). [edit]. ...
en.wikipedia.org/wiki/Secret_sharing - 25k - Cached - Similar pages

How to **share** a **secret**
... Feedback Report a problem Satisfaction survey. How to **share** a **secret**. ... Author, Adi **Shamir**, Massachusetts Institute of Technology, Cambridge. ...
portal.acm.org/citation.cfm?id=359176 - Similar pages

Shamir's Secret Sharing
How to **Share** a **Secret**. Adi **Shamir** Massachusetts Institute of Technology In this paper we show how to divide data D into n pieces ...
www.cs.tau.ac.il/~bchor/Shamir.html - 11k - Cached - Similar pages

2.2.1.1 **Shamir Secret Sharing**
next up previous contents Next: 2.2.1.2 Sum **Secret Sharing** Up: 2.2.1 **Secret Sharing** Previous: 2.2.1 **Secret Sharing**. 2.2.1.1 **Shamir Secret Sharing**. ...
www.digitas.harvard.edu/~ken/cs262/election/node7.html - 8k - Cached - Similar pages

Bibliography on **Secret Sharing** Schemes
... Theory B 56 (1992), 69-73. A. **Shamir**, How to **share** a **secret**, Communications of the ACM 22 (1979), 612-613. SP Shieh and HM Sun, On ...
www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html - 45k - Cached - Similar pages

RSA Security - 3.6.12 What are some **secret sharing** schemes?
... Shamir's secret sharing scheme [Sha79] is a threshold scheme based on polynomial interpolation. To allow any m out of n people to ...
www.rsasecurity.com/rsalabs/node.asp?id=2259 - 18k - Cached - Similar pages

Gooooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Get the Google Toolbar. Download Now - About Toolbar



how to share a secret shamir Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2005 Google

Google Web Images Groups News Froogle Local[New!] more »
secret key sharing Search Advanced Search Preferences

Web

Results **11** - **20** of about **1,590,000** for **secret key sharing**. (**0.13** seconds)

Secret-sharing for GPG?
... Next message: **Secret-sharing** for GPG? ... process would have to work on **keys** - invocation would be something like: (1) - fragment this **secret key** (needs passphrase ...
lists.gnupg.org/pipermail/ gnupg-devel/1998-December/015031.html - 4k - Cached - Similar pages

Secret-sharing for GPG?
... and rip off their employers all the time.) > > There are ways to split **keys** (mathematically) that allow **key sharing** > > with no central **secret key** storage. ...
lists.gnupg.org/pipermail/ gnupg-devel/1998-December/015030.html - 7k - Cached - Similar pages
[More results from lists.gnupg.org]

[PDF] **Secret Sharing** and Splitting
File Format: PDF/Adobe Acrobat - View as HTML
... the **secret key**. 1 Keywords applied cryptography, cryptology, XOR, **secret sharing**, **secret** splitting, **key** management 2 Introduction ...
www.cse.nd.edu/~cseprog/proj02/cryptogrophy/final.pdf - Similar pages

Secret sharing - Wikipedia, the free encyclopedia
... So far so good, but who gets a copy of the **key**? An appropriate **secret sharing** scheme could protect access to the decrypted file (the formula/combination) by ...
en.wikipedia.org/wiki/Secret_**sharing** - 25k - Cached - Similar pages

RSA Security - 3.6.1 What is Diffie-Hellman?
... Suppose Alice and Bob want to agree on a **shared secret key** using the Diffie-Hellman **key** agreement protocol. They proceed as follows ...
www.rsasecurity.com/rsalabs/node.asp?id=2248 - 18k - Cached - Similar pages
[More results from www.rsasecurity.com]

Shamir's **Secret Sharing**
How to **Share** a **Secret**. ... of k - 1 pieces reveals absolutely no information about D. This technique enables the construction of robust **key** management schemes or ...
szabo.best.vwh.net/**secret**.html - 11k - Cached - Similar pages

Key Distribution
... B) and with the use of H(a,b) as the **shared secret key**. Another form involves a table with groups of three entries in the form: ...
home.ecn.ab.ca/~jsavard/crypto/mi060710.htm - 14k - Cached - Similar pages

Tales From The Crypto - Part I
... You and recipient **share** a **secret key**. ... It eliminates the need to exchange or **share** a **secret key** between every two parties wishing to communicate privately. ...
searchsecurity.techtarget.com/ tip/1,289483,sid14_gci299975,00.html - 41k - Cached - Similar pages

The Protection of Your **Secret Key**
... there is no way to decrypt or sign a message without using the **secret key**. ... Bell Labs, Morristown, New Jersey) on their workstations at night **sharing** the work ...
senderek.de/security/**secret**-**key**.protection.html - 98k - Cached - Similar pages

[PDF] **Secret sharing** with access structures in a hierarchy - Advanced ...
File Format: PDF/Adobe Acrobat - View as HTML
... u i,|A| such that ,1 ,1 ,2 ,2 , , ... i i i i i iAiA K su su su (9) For conditional **secret sharing** based upon the discrete logarithm, we assume a **secret key**
islab.cis.thu.edu.tw/files/teacher/194.pdf - Similar pages

Gooooooooooogle
Result Page: **Previous** 1 2 3 4 5 6 7 8 9 10 11 **Next**

secret key sharing Search

Search within results | Language Tools | Search Tips

Google Home - Advertising Programs - Business Solutions - About Google

©2005 Google

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

key splitting

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **key splitting** Found **30,273** of **150,138**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

21 LH*—a scalable, distributed data structure

Witold Litwin, Marie-Anna Neimat, Donovan A. Schneider

December 1996 **ACM Transactions on Database Systems (TODS)**, Volume 21 Issue 4

Full text available: pdf(780.53 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We present a scalable distributed data structure called LH*. LH* generalizes Linear Hashing (LH) to distributed RAM and disk files. An LH* file can be created from records with primary keys, or objects with OIDs, provided by any number of distributed and autonomous clients. It does not require a central directory, and grows gracefully, through splits of one bucket at a time, to virtually any number of servers. The number of messages per random insertion is one in general, and three in the w ...

Keywords: algorithms, data structures, distributed access methods, extensible hashing, linear hashing

22 Prefix B-trees

Rudolf Bayer, Karl Unterauer

March 1977 **ACM Transactions on Database Systems (TODS)**, Volume 2 Issue 1

Full text available: pdf(1.13 MB) Additional Information: full citation, abstract, references, citings, index terms

Two modifications of B-trees are described, simple prefix B-trees and prefix B-trees. Both store only parts of keys, namely prefixes, in the index part of a B*-tree. In simple prefix B-trees those prefixes are selected carefully to minimize their length. In prefix B-trees the prefixes need not be fully stored, but are reconstructed as the tree is searched. Prefix
Keywords: B-trees, key compression, multiway search trees

23 New techniques for security and reliability enhancement in embedded systems: Low energy security optimization in embedded cryptographic systems

Catherine H. Gebotys

September 2004 **Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis**

Full text available: pdf(252.09 KB) Additional Information: full citation, abstract, references, index terms

Future embedded and wireless devices will be increasingly powerful supporting many

applications including one of the most crucial, security. Although many wireless and embedded devices offer more resistance to bus probing attacks due to their compact size, susceptibility to power/electromagnetic attacks must be analyzed. This paper presents optimized synthesis of new low energy masking countermeasures into cryptographic software. In particular a model for key masking with the objective of minimi ...

Keywords: countermeasure, embedded, power analysis, smart cards

24 System-level exploration of association table implementations in telecom network applications

Ch. Ykman-Couvreur, J. Lambrecht, A. Van Der Togt, F. Catthoor, H. De Man

November 2002 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 1 Issue 1

Full text available: pdf(375.20 KB) Additional Information: full citation, abstract, references, index terms

We present a new exploration and optimization method at the system level to select customized implementations for dynamic data sets, as encountered in telecom network, database, and multimedia applications. Our method fits in the context of embedded system synthesis for such applications, and enables to further raise the abstraction level of the initial specification, where dynamic data sets can be specified without low-level details. Our method is suited for hardware and software implementation ...

Keywords: System-level exploration, memory management

25 Concurrency in linear hashing

Carla Schlatter Ellis

June 1987 **ACM Transactions on Database Systems (TODS)**, Volume 12 Issue 2

Full text available: pdf(1.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Concurrent access to complex shared data structures, particularly structures useful as database indices, has long been of interest in the database community. In dynamic databases, tree structures such as B-trees have been used as indices because of their ability to handle growth; whereas hashing has been used for fast access in relatively static databases. Recently, a number of techniques for dynamic hashing have appeared. They address the major deficiency of traditional hashing when applie ...

26 Splitting long or ill-formed input for robust spoken-language translation

Osamu Furuse, Setsuo Yamada, Kazuhide Yamamoto

August 1998

Full text available: pdf(639.58 KB) Publisher Site Additional Information: full citation, abstract, references

This paper proposes an input-splitting method for translating spoken-language which includes many long or ill-formed expressions. The proposed method splits input into well-balanced translation units based on a semantic distance calculation. The splitting is performed during left-to-right parsing, and does not degrade translation efficiency. The complete translation result is formed by concatenating the partial translation results of each split unit. The proposed method can be incorporated into ...

27 Efficient locking for concurrent operations on B-trees

Philip L. Lehman, s. Bing Yao

December 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 4

Additional Information:

Full text available: pdf(1.40 MB) full citation, abstract, references, citings, index terms

The B-tree and its variants have been found to be highly useful (both theoretically and in practice) for storing large amounts of information, especially on secondary storage devices. We examine the problem of overcoming the inherent difficulty of concurrent operations on such structures, using a practical storage model. A single additional "link" pointer in each node allows a process to easily recover from tree modifications performed by other concurrent processes. Our solution ...

Keywords: B-tree, concurrenct algorithms, concurrency controls, consistencey, correctness, data structures, database, index organizations, locking protocols, multiway search trees

28 Physical storage structures: Experiments with B-tree reorganization
Ehud Gudes, Shalom Tsur
May 1980 **Proceedings of the 1980 ACM SIGMOD international conference on Management of data**
Full text available: pdf(567.90 KB) Additional Information: full citation, abstract, references, citings

B-trees are a commonly used data structure for indexed access to files and databases. Among the desirable properties of B-trees is the fact that they are dynamically rebalanced after each insertion and deletion operation and therefore need not be reorganized as other static access structures e.g., ISAM. Despite the fact that B-trees are dynamically balanced we demonstrate that operational conditions exist under which it pays off to explicitly reorganize B-trees. The rationale being that by expli ...

29 Ubiquitous B-Tree
Douglas Comer
June 1979 **ACM Computing Surveys (CSUR)**, Volume 11 Issue 2
Full text available: pdf(1.37 MB) Additional Information: full citation, references, citings, index terms

30 Fast algorithms for k-shredders and k-node connectivity augmentation (extended abstract)
Joseph Cheriyan, Ramakrishna Thurimella
July 1996 **Proceedings of the twenty-eighth annual ACM symposium on Theory of computing**
Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

31 Performance of B+ tree concurrency control algorithms
V. Srinivasan, Michael J. Carey
October 1993 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 2 Issue 4
Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, references, citings

A number of algorithms have been proposed to access B+-trees concurrently, but they are not well understood. In this article, we study the performance of various B+-tree concurrency control algorithms using a detailed simulation model of B+-tree operations in a centralized DBMS. Our study covers a wide range of data contention situations and resource conditions. In addition, based on the performance of the set of B+-tree concurrency control algorithms, ...

Keywords: B+-tree structures, data contention, lock modes, performance, resource

conditions, simulation models, workload parameters

32 The hB $^\Pi$-tree: a multi-attribute index supporting concurrency, recovery and node consolidation

Georgios Evangelidis, David Lomet, Betty Salzberg

February 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 1

Full text available: pdf(314.15 KB) Additional Information: full citation, abstract, index terms

We propose a new multi-attribute index. Our approach combines the hB-tree, a multi-attribute index, and the Π-tree, an abstract index which offers efficient concurrency and recovery methods. We call the resulting method the hB $^\Pi$-tree. We describe several versions of the hB $^\Pi$-tree, each using a different node-splitting and index-term-posting algorithm. We also describe a new node deletion algorithm. We have implemented all the versions of the hB $^\Pi$-tree. Our performance results ...

Keywords: Concurrency, Multi-attribute index, Node consolidation, Recovery

33 Performance analysis of file organizations that use multibucket data leaves with partial expansions

Gabriel Matsliach

March 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 1

Full text available: pdf(1.57 MB) Additional Information: full citation, abstract, references, index terms, review

We present an exact performance analysis, under random insertions, of file organizations that use multibucket data leaves and perform partial expansions before splitting. We evaluate the expected disk space utilization of the file and show how the expected search and insert costs can be estimated. The analytical results are confirmed by simulations. The analysis can be used to investigate both the dynamic and the asymptotic behaviors.

Keywords: bounded disorder files, multibucket data leaves, partial expansion, performance analysis, search structures

34 Dynamic hashing schemes

R. J. Enbody, H. C. Du

July 1988 **ACM Computing Surveys (CSUR)**, Volume 20 Issue 2

Full text available: pdf(2.52 MB) Additional Information: full citation, abstract, references, citings, index terms

A new type of dynamic file access called dynamic hashing has recently emerged. It promises the flexibility of handling dynamic files while preserving the fast access times expected from hashing. Such a fast, dynamic file access scheme is needed to support modern database systems. This paper surveys dynamic file access scheme is needed to support modern database systems. This paper surveys dynamic hashing schemes and examines their critical design issues.

Keywords: dynamic hashing

35 Analysis of dynamic hashing with deferred splitting

Eugene Veklerov

March 1985 **ACM Transactions on Database Systems (TODS)**, Volume 10 Issue 1

Full text available: pdf(433.18 KB) Additional Information: full citation, abstract, references, citings, index terms

Dynamic hashing with deferred splitting is a file organization scheme which increases storage utilization, as compared to standard dynamic hashing. In this scheme, splitting of a bucket is deferred if the bucket is full but its brother can accommodate new records. The performance of the scheme is analyzed. In a typical case the expected storage utilization increases from 69 to 76 percent.

36 A scalable multicast key management scheme for heterogeneous wireless networks

Yan Sun, Wade Trappe, K. J. Ray Liu

August 2004 **IEEE/ACM Transactions on Networking (TON)**, Volume 12 Issue 4

Full text available: pdf(825.80 KB) Additional Information: full citation, abstract, references, index terms

Secure multicast applications require key management that provides access control. In wireless networks, where the error rate is high and the bandwidth is limited, the design of key management schemes should place emphasis on reducing the communication burden associated with key updating. A communication-efficient class of key management schemes is those that employ a tree hierarchy. However, these tree-based key management schemes do not exploit issues related to the delivery of keying informat ...

Keywords: communication system security, multicast key management, secure handoff, wireless multicast

37 Online feedback-directed optimization of Java

Matthew Arnold, Michael Hind, Barbara G. Ryder

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 37 Issue 11

Full text available: pdf(463.00 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the implementation of an online feedback-directed optimization system. The system is fully automatic; it requires no prior (offline) profiling run. It uses a previously developed low-overhead instrumentation sampling framework to collect control flow graph edge profiles. This profile information is used to drive several traditional optimizations, as well as a novel algorithm for performing feedback-directed control flow graph node splitting. We empirically evaluate this syst ...

Keywords: adaptive optimization, dynamic optimization, online algorithms, virtual machines

38 A multikey hashing scheme using predicate trees

Patrick Valduriez, Yann Viemont

June 1984 **ACM SIGMOD Record , Proceedings of the 1984 ACM SIGMOD international conference on Management of data**, Volume 14 Issue 2

Full text available: pdf(815.03 KB) Additional Information: full citation, abstract, references, citings

A new method for multikey access suitable for dynamic files is proposed that transforms multiple key values into a logical address This method is based on a new structure, called predicate tree, that represents the function applied to several keys A predicate tree permits to specify in a unified way various hashing schemes by allowing for different definitions of predicates A logical address qualifies a space partition of a file according to its predicate tree This address is seen as a single ke ...

39 Concurrency and recovery for index trees

David Lomet, Betty Salzberg

August 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 3

Full text available: pdf(168.36 KB) Additional Information: full citation, abstract, citings, index terms

Although many suggestions have been made for concurrency in B $^+$-trees, few of these have considered recovery as well. We describe an approach which provides high concurrency while preserving well-formed trees across system crashes. Our approach works for a class of index trees that is a generalization of the B $^{\rm link}$-tree. This class includes some multi-attribute indexes and temporal indexes. Structural changes in an index tree are decomposed into a sequence of atomic actions, each one ...

Keywords: Access methods, B-trees, Concurrency, Indexing, Recovery

40 Some optimal inapproximability results

Johan Håstad

July 2001 **Journal of the ACM (JACM)**, Volume 48 Issue 4

Full text available: pdf(556.67 KB) Additional Information: full citation, abstract, references, citings, index terms

We prove optimal, up to an arbitrary $\varepsilon > 0$, inapproximability results for Max-E k-Sat for $k \geq 3$, maximizing the number of satisfied linear equations in an over-determined system of linear equations modulo a prime p and Set Splitting. As a consequence of these results we get improved lower bounds for the efficient approximability of many optimization problems studied previously. In particular, for Max-E2-Sat, Max-Cut, Max-di-Cut, and Vertex cover.

Keywords: Inapproximability, NP-hard optimization problems, linear equations, max-sat, probabilistically checkable proofs

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore RELEASE 1.8

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links

» Se

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced
- CrossRef

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **14** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Tracing traitors**
Chor, B.; Fiat, A.; Naor, M.; Pinkas, B.;
Information Theory, IEEE Transactions on ,Volume: 46 , Issue: 3 , May 2000
Pages:893 - 910

[Abstract] [PDF Full-Text (356KB)] **IEEE JNL**

2 **Secret sharing with public reconstruction**
Beimel, A.; Chor, B.;
Information Theory, IEEE Transactions on ,Volume: 44 , Issue: 5 , Sept. 199
Pages:1887 - 1896

[Abstract] [PDF Full-Text (380KB)] **IEEE JNL**

3 **Secret Sharing With Public Reconstruction**
Beimel, A.; Chor, B.;
Information Theory, IEEE Transactions on ,Volume: 44 , Issue: 5 , Sept. 199
Pages:1887 - 1896

[Abstract] [PDF Full-Text (380KB)] **IEEE JNL**

4 **Communication in key distribution schemes**
Beimel, A.; Chor, B.;
Information Theory, IEEE Transactions on ,Volume: 42 , Issue: 1 , Jan. 1996
Pages:19 - 28

[Abstract] [PDF Full-Text (960KB)] **IEEE JNL**

5 **Universally ideal secret-sharing schemes**
Beimel, A.; Chor, B.;
Information Theory, IEEE Transactions on ,Volume: 40 , Issue: 3 , May 1994
Pages:786 - 794

[Abstract] [PDF Full-Text (816KB)] **IEEE JNL**

6 **Privacy, additional information and communication**

Bar-Yehuda, R.; Chor, B.; Kushilevitz, E.; Orlitsky, A.;
Information Theory, IEEE Transactions on ,Volume: 39 , Issue: 6 , Nov. 1993
Pages:1930 - 1943

[Abstract] [PDF Full-Text (1140KB)] **IEEE JNL**

7 **A knapsack-type public key cryptosystem based on arithmetic in fini fields**
Chor, B.; Rivest, R.L.;
Information Theory, IEEE Transactions on ,Volume: 34 , Issue: 5 , Sept. 198
Pages:901 - 909

[Abstract] [PDF Full-Text (756KB)] **IEEE JNL**

8 **Tight lower bounds for certain parameterized NP-hard problems**
Chen, J.; Chor, B.; Fellows, M.; Xiuzhen Huang; Juedes, D.; Kanj, I.; Ge Xia;
Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on , 21-24 June 2004
Pages:150 - 160

[Abstract] [PDF Full-Text (404KB)] **IEEE CNF**

9 **RSA/Rabin Bits are 1/2 + 1 Poly (Log N) Secure**
Alexi, W.; Chor, B.; Goldreich, O.; Schnorr, C.P.;
Foundations of Computer Science, 1984. 25th Annual Symposium on , Octobe 26, 1984
Pages:449 - 457

[Abstract] [PDF Full-Text (648KB)] **IEEE CNF**

10 **Private information retrieval**
Chor, B.; Goldreich, O.; Kushilevitz, E.; Sudan, M.;
Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposiu on , 23-25 Oct. 1995
Pages:41 - 50

[Abstract] [PDF Full-Text (948KB)] **IEEE CNF**

11 **Private computations over the integers**
Chor, B.; Gereb-Graus, M.; Kushilevitz, E.;
Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposiu on , 22-24 Oct. 1990
Pages:335 - 344 vol.1

[Abstract] [PDF Full-Text (656KB)] **IEEE CNF**

12 **Privacy, additional information, and communication**
Bar-Yehuda, R.; Chor, B.; Kushilevitz, E.;
Structure in Complexity Theory Conference, 1990, Proceedings., Fifth Annual 11 July 1990
Pages:55 - 65

[Abstract] [PDF Full-Text (720KB)] **IEEE CNF**

13 **Solvability in asynchronous environments**
Chor, B.; Moscovici, L.;
Foundations of Computer Science, 1989., 30th Annual Symposium on , 30 Oct Nov. 1989
Pages:422 - 427

[Abstract] [PDF Full-Text (440KB)] **IEEE CNF**

14 **On the theory of average case complexity**
Ben-David, S.; Chor, B.; Goldreich, O.; Luby, M.;
Structure in Complexity Theory Conference, 1989. Proceedings., Fourth Annual , 19-22 June 1989
Pages:36

[Abstract] [PDF Full-Text (32KB)] **IEEE CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Copyright © 2004 IEEE — All rights reserved